

JSP: Java Server Page

IUT Lyon 1

Yosra ZGUIRA

yosra.zguira@insa-lyon.fr

2016 - 2017

C'est quoi le JSP (Java Server Page)?

- Le langage JSP est un langage de scripts composé à la fois :
 - ✓ de balises **HTML**
 - ✓ et d'instructions provenant du langage de programmation **Java**.
- On peut mettre des pages .jsp partout où on met des pages HTML.
- Elles sont converties "à la volée" en servlet par le moteur de JSP.
- Pour que le serveur puisse différencier le code HTML du **code JSP**, il est nécessaire d'entourer les instructions JSP par des éléments de script spécifiques, appelés **étiquettes** ou encore **balises JSP**.



Ecrire un script JSP demande donc d'acquérir une bonne connaissance du code HTML et d'apprendre à programmer en Java.

Présentation des JSP (1/3)

- **Avec les Servlets, il est facile de :**
 - ✓ lire des formulaires
 - ✓ lire les entêtes **HTTP**
 - ✓ envoyer les entêtes réponses, les codes d'état, les pages réponses
 - ✓ partager les données entre Servlets
 - ✓ mémoriser des informations entre les requêtes
- **Mais il est pénible de :**
 - ✓ gérer les instructions **println** de génération d'**HTML**

Présentation des JSP (2/3)

- **Principe**

- ✓ utilisation du HTML classique pour la majorité de la page,
- ✓ Ajout du code Java marqué par des balises spéciales,
- ✓ la page JSP est traduite par le serveur en Servlet une seule fois,
- ✓ la Servlet obtenue est invoquée à chaque requête.

- **Exemple d'une portion de page JSP**

```
<H3>Aujourd'hui, nous sommes le:</h3>
```

```
<|> <%= (new java.util.Date()) %> </|>
```

- ✓ **Résultat**

✗ **Aujourd'hui, nous somme le *Thu Oct 11:08:53 CEST 2016***

Présentation des JSP (3/3)

- **Avantage des JSP:**

- ✓ écriture directe du **HTML**
- ✓ possibilité d'utiliser les outils **HTML** classique
- ✓ possibilité de séparer la présentation réalisée en **HTML** et le traitement réalisé en **Java**
- ✓ possibilité de modularité et de réutilisation en utilisant des **beans** dans la partie **Java**
- ✓ appel des pages **JSP** comme des pages **HTML** standards, par utilisation d'une **URL** avec un nom de fichier avec une extension **.jsp**

✗ **http:\\localhost:8080\\MaWebApp\\xxxx.jsp**

Radiographie d'un script JSP (1/16)

- Considérons un exemple de code source d'un premier programme JSP qui affiche la liste suivante des réalisateurs et des films : L. Besson, le grand bleu; F. Truffaut, Jules et Jim; J.-J. Annaud, la guerre du feu; J.-J. Beineix, 37°2 le matin.
- Cette liste doit s'afficher sous la forme d'un tableau à deux colonnes, la première pour les réalisateurs, la seconde pour les films réalisés par ces auteurs.

Réalisateurs	Films
L. Besson	Le grand bleu
F. Truffaut	Jules et Jim
J-J Annaud	La guerre du feu
J-J Beineix	37°2 le matin

Radiographie d'un script JSP (2/16)

```
<!-- Commentaire visible : le premier JSP étudié -- >
<html><head> <title> Un premier exemple de JSP </title> </head>
<!-- définition des informations globales de la page -->
<% @ page contentType="text/html" %>
<%-- déclaration de variables -- %>
<%!
String IFilm [] = { "le grand bleu", "Jules et Jim", "La guerre du feu", "37°2 le matin" };
String IRealisateur [] = {"L. Besson", "F. Truffaut", "J.-J. Annaud", "J.-J. Beineix" };
String IAffiche="";
int i;
%>
<body>
<%
IAffiche = "<table border=1>";
IAffiche+= "<TR><TH> Réalisateur </TH><TH>Film</TH></TR>";
for (i=0; i< 4; i++)
{IAffiche+= "<TR><TD>" + IRealisateur[i] + "</TD>";
  IAffiche+= "<TD>" + IFilm[i] + "</TD></TR>";
}
IAffiche+= "</table>";
%>
<%= IAffiche %>
</body></html>
```

Radiographie d'un script JSP (3/16)

- **Un script JSP contient:**

- ✓ Des balises HTML:

- ✗ Exemples: `<html>`, `<head>`, `<title>`, `<body>`, `</html>`, `</head>`, `</title>`, `</body>`, etc.

- ✓ Des Balises JSP:

- ✗ Les balises de commentaire,

- ✗ Les balises de déclaration,

- ✗ Les balises de scriptlet,

- ✗ Les balises d'expression,

- ✗ Les balises de directives.

- ✓ Des Variables Prédéfinis

Radiographie d'un script JSP (4/16)

- Les balises de commentaire:

- ✓ Balises de commentaires visibles (code source HTML):

- ✗ Les balises `<!-- -->` sont utilisées pour placer un commentaire dans la page HTML générée par le serveur. Ainsi le commentaire :

- `<!-- Commentaire visible : le premier JSP étudié -->`

- ✗ est placé dans le code de la page HTML résultante de la façon suivante :

- `<!-- Commentaire visible : le premier JSP étudié -->`

Radiographie d'un script JSP (5/16)

- Les balises de commentaire (suite):

- ✓ Balises de commentaires cachés ou commentaire JSP :

- ✗ Elles ne sont pas pris en compte par l'interpréteur JSP. Par conséquent les commentaires n'ont pas à être communiqués à l'internaute.

- ✗ Les balises `<%-- --%>` permettent de commenter tout simplement le code JSP, les commentaires n'étant pas visible sur le code source de la page HTML résultante:

`<%-- déclaration de variables -- %>`

Radiographie d'un script JSP (6/16)

- Les balises de déclaration:

- ✓ Le langage JSP propose, comme tout langage de programmation, la possibilité de déclarer des variables.
- ✓ Cette déclaration se fait à l'aide des balises `<%! %>`.
- ✓ Exemple :

```
<%!  
String IFilm [] = { "le grand bleu", "Jules et Jim", "La guerre du feu", "37°2 le matin" };  
String IRealisateur [] = {"L. Besson", "F. Truffaut", "J.-J. Annaud", "J.-J. Beineix" };  
String IAffiche="";  
int i;  
%>
```

- ✗ Déclaration de deux tableaux de chaîne de caractères nommés `IRealisateur` et `IFilm`, une chaîne de caractères vide nommée `IAffiche` et une variable `i` pour stocker un entier.

Radiographie d'un script JSP (7/16)

- Les balises de scriptlets:

- ✓ Le balises `<% %>` permettent d'insérer à l'intérieur d'une page HTML toutes les instructions de type déclaration, test, boucle, et affichage décrites par le langage Java.
- ✓ Exemple :

```
<%  
    lAffiche = "<table border=1>";  
    lAffcihe+= "<TR><TH> +Réalisateur </TH><TH>Film</TH></TR>";  
    for (i=0; i< 4; i++)  
    { lAffcihe+= "<TR><TD>"+lRealisateur[i]+"</TD>";  
      lAffcihe+= "<TD>"+lFilm[i]+"</TD></TR>";  
    }  
    lAffcihe+= "</table>";  
%>
```

Radiographie d'un script JSP (8/16)

- Les balises d'expression:

- ✓ La chaîne `lAffiche` contient les balises `HTML` ainsi que les noms des réalisateurs et des films. Cette liste de mots est stockée en mémoire mais n'est pas encore affichée par le navigateur.
- ✓ Pour placer le contenu de la chaîne dans le flux de la réponse HTTP du serveur, nous devons utiliser une balise d'expression comme suit :

`<% = lAffiche %>`

- ✓ En insérant la chaîne `lAffiche` entre les balises `<%=` et `%>`, le contenu de la chaîne est envoyé au navigateur qui affiche un tableau.

NB: Une balise d'expression ne contient jamais de point-virgule.

Radiographie d'un script JSP (9/16)

- Les balises de directive:

- ✓ Format :

- <%@ directive attribut="valeur" %>

- ✓ Exemple de directives possibles :

- ✗ **page: informations relatives à la page**

- > <%@ page import="java.util.*, java.net.*" %>

- > <%@ page contentType="text/plain" %>

- > <%@ page session="true|false " %>

- ✗ **include: fichiers à inclure littéralement au moment de la traduction de la page**

- > <%@ include file="chemin relatif du fichier" %>

- > <%@ include file="/toto.html" %>

Radiographie d'un script JSP (10/16)

- Les variables prédéfinis:

- ✓ **request**

- ✗ **HttpServletRequest**: (1er argument de service/doGet/doPost)

- ✓ **response**

- ✗ **HttpServletResponse**: (2ème argument de service/doGet/doPost)

- ✓ **out**

- ✗ Un **Writer**, utilisé pour envoyé des informations au client

- ✓ **session**

- ✗ **HttpSession**: associée avec le request (si n'est pas annulée par l'attribut session de la directive page)

- ✓ **application**

- ✗ **ServletContext**: obtenu avec `getServletContext()`.

Radiographie d'un script JSP (11/16)

- **Exemple1:**

- Ecrire le script JSP `receptionPasswd.jsp` qui permet d'afficher un mot de passe saisi à l'aide d'un formulaire décrit dans le fichier `formulaire.html`.

formulaire.html

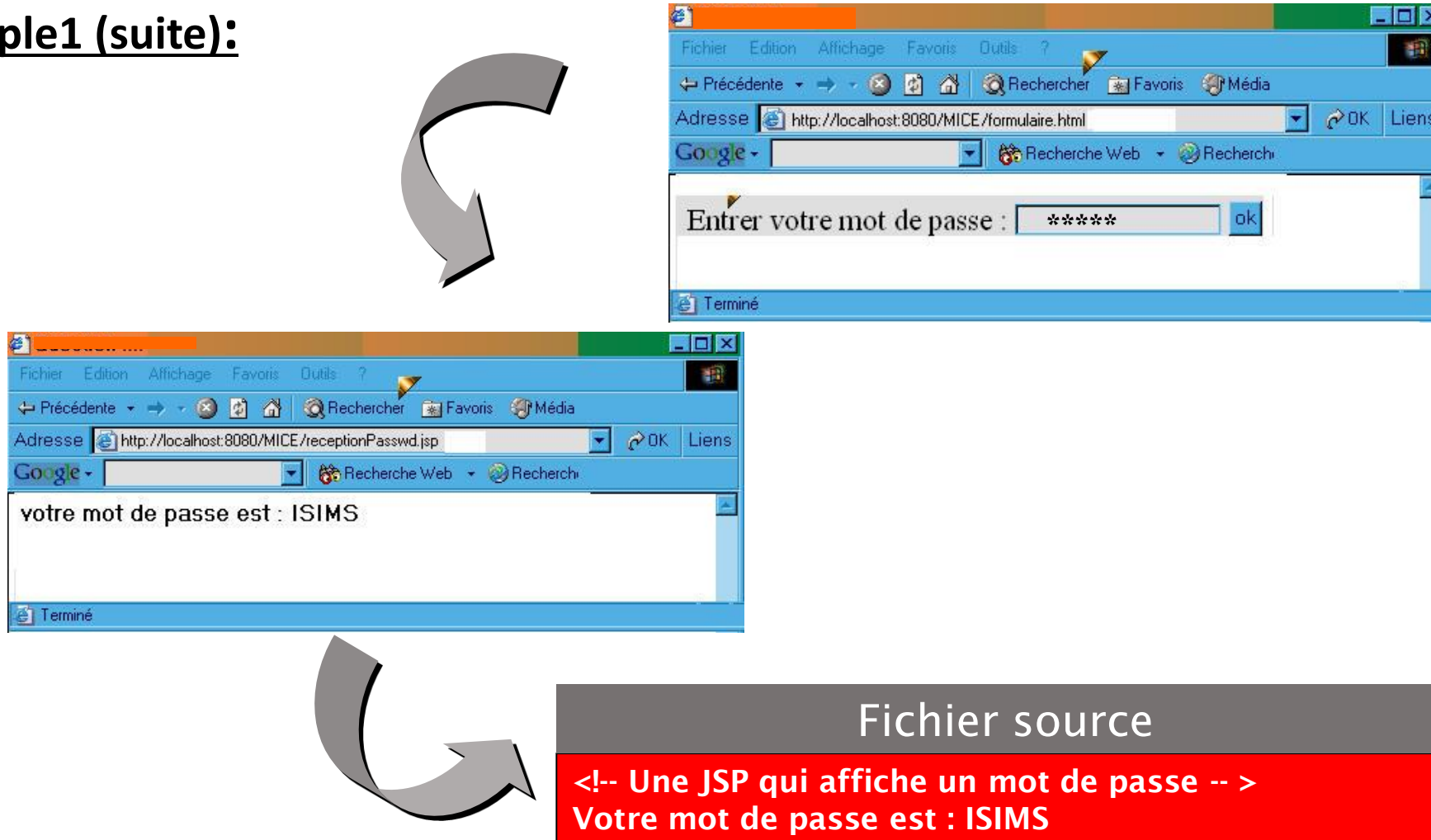
```
<Form name=ident action=http://localhost:8080/MaWebApp/receptionPasswd.jsp method=POST>
  Entrer votre mot de passe : <input type="password" name="valeur" maxlength="8">
  <input type="submit" value="ok">
</form>
</body>
</html>
```

receptionPasswd.jsp

```
<!-- Une JSP qui affiche un mot de passe -- >
<%
  String motdepasse = request.getParameter("valeur");
  out.println( "votre mot de passe est : "+ motdepasse);
%>
```

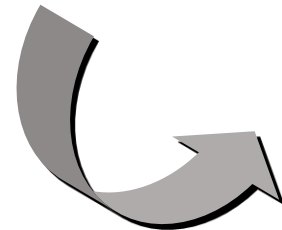
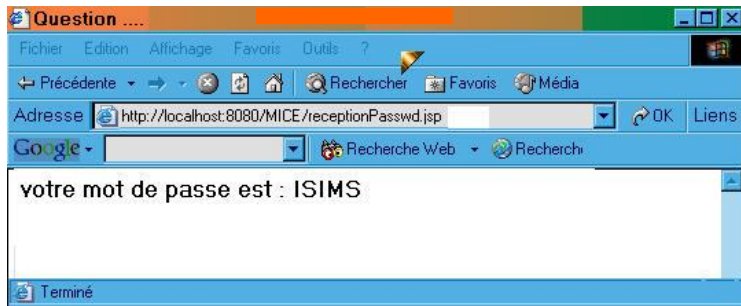
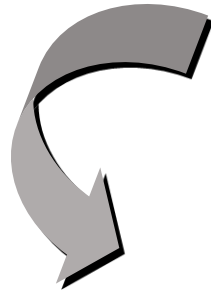

Radiographie d'un script JSP (12/16)

- Exemple1 (suite):



Radiographie d'un script JSP (13/16)

- Exemple2:



receptionPasswd.jsp

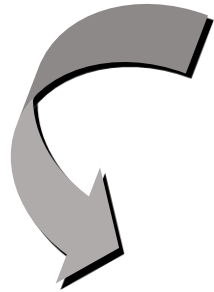
```
<!-- Une JSP qui affiche un mot de passe -- >
<% String motdepasse = request.getParameter("valeur"); %>
<html>
<Head>
<title> Afficahge de mot de passe </title>
<%
    out.println("<body> votre mot de passe est : " + motdepasse+"</body>");
%>
</html>
```

Fichier source

```
<!-- Une JSP qui affiche un mot de passe -- >
<html>
<Head>
<title> Afficahge de mot de passe </title>
<body> votre mot de passe est : ISIMS</body>
</html>
```

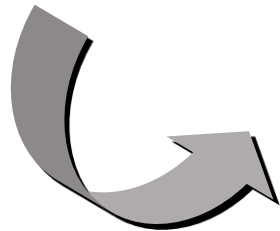
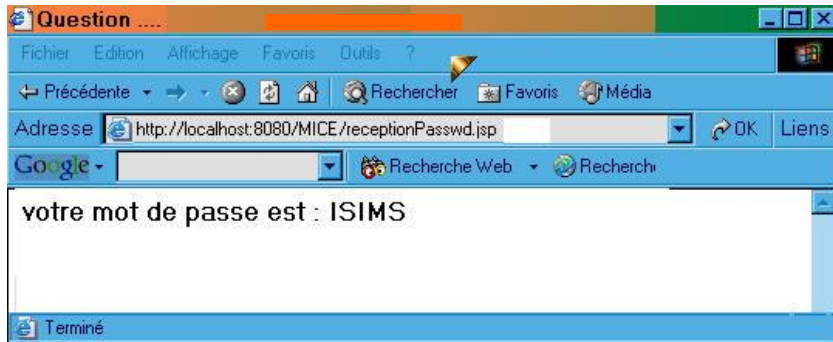
Radiographie d'un script JSP (14/16)

- Exemple3:



receptionPasswd.jsp

```
<!-- Une JSP qui affiche un mot de passe -- >
<% String motdepasse = request.getParameter("valeur"); %>
<html>
<Head>
<title> Afficahge de mot de passe </title>
<body> votre mot de passe est : <%= motdepasse %> </body>
</html>
```

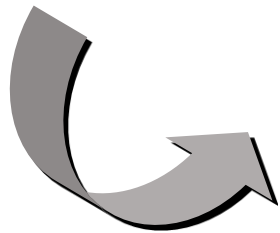
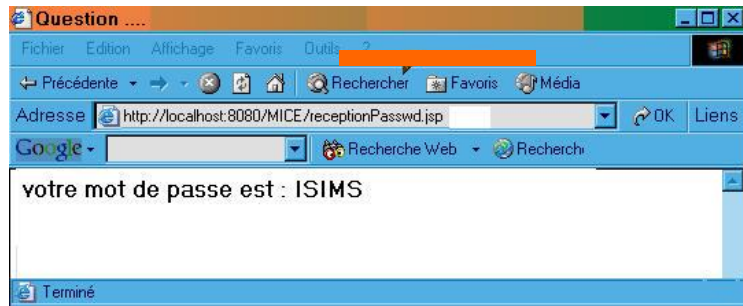
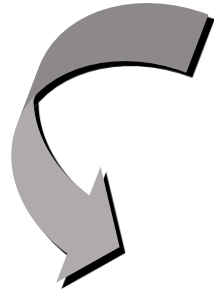


Fichier source

```
<!-- Une JSP qui affiche un mot de passe -- >
<html>
<Head>
<title> Afficahge de mot de passe </title>
<body> votre mot de passe est : ISIMS </body>
</html>
```

Radiographie d'un script JSP (15/16)

- Exemple4:



receptionPasswd.jsp

```
<%@ page import="java.util.*" %>
<!-- Une JSP qui affiche un mot de passe -- >
<!-- Script JSP Créé le : <%=new Date()%> -->
<% String motdepasse = request.getParameter("valeur"); %>
<html>
<Head>
<title> Afficahge de mot de passe </title>
<body> votre mot de passe est : <%= motdepasse %></body>
</html>
```

Fichier source

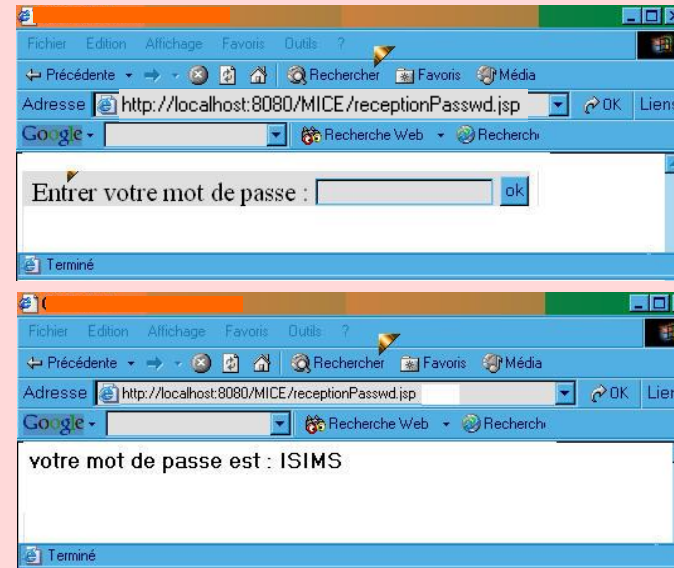
```
<!-- Une JSP qui affiche un mot de passe -- >
<!-- Script JSP Créé le : Mon Oct 25 01:46:09 CEST 2004-->
<html>
<Head>
<title> Afficahge de mot de passe </title>
<body> votre mot de passe est : ISIMS </body>
</html>
```

Radiographie d'un script JSP (16/16)

- Exemple5: AffichePasswd.jsp = receptionPasswd.jsp + formulaire.html

AffichePasswd.jsp

```
<%@ page import="java.util.*" %>
<!-- Une JSP qui affiche un mot de passe -->
<!-- Script JSP Créé le : <%=new Date()%>-->
<html><head>
<title> Afficahge de mot de passe </title>
<body>
<% String motdepasse = request.getParameter("valeur");
    if (motdepasse !=null)
    {
<%
votre mot de passe est : <%=motdepasse%>
<%
    }
    else
    {
<%
<form name=ident action=http://localhost:8080/MICE/AffichePasswd.jsp method=POST>
    Entrer votre mot de passe : <input type ="password" name="valeur" maxlength="8">
    <input type="submit" value="ok">
    </form>
<% } %>
</body></head></html>
```



Du JSP à la Servlet (1/3)

- Lorsque le serveur reçoit une requête du type <http://localhost/MaWebApp/MaJSP.jsp>, le serveur lance un **interpréteur de commande JSP** qui convertit l'ensemble des instructions placées dans le fichier **MaJSP.jsp** en un programme **Java** de type **Servlet**.
- Une fois la Servlet est entièrement écrite, le serveur la compile par l'intermédiaire de la **JVM**, la charge en mémoire et l'exécute automatiquement afin de générer un résultat sous la forme d'une page HTML.

Du JSP à la Servlet (2/3)

- **Principe de traduction du JSP en Servlet:**

- ✓ Les variables et les méthodes déclarées dans une balise `<%! ... %>` sont traduites en variables et méthodes d'instance.
- ✓ Les variables déclarées à l'intérieur d'une scriptlet sont transformées en variables locales d'une méthode appelé `_jspService()`.
- ✓ Les expressions JSP du type `<%=... %>` sont placées dans un flux de sortie de type `out.println()`.
- ✓ Les commentaires JSP sont ignorés.
- ✓ Les balises HTML sont disposées à l'intérieur de la méthode `_jspService()` dans leur ordre d'arrivée et sont placées à l'intérieur d'un flux d'écriture de type `out.println()`.
- ✓ Le code Java défini à l'intérieur des scriptlets est placé tel quel à l'intérieur de la méthode `_jspService()`.

Du JSP à la Servlet (3/3)

- **Exemple de script JSP:**

```
<H3>Aujourd'hui, nous sommes le:</h3>  
<|> <%= (new java.util.Date()) %> </|>
```

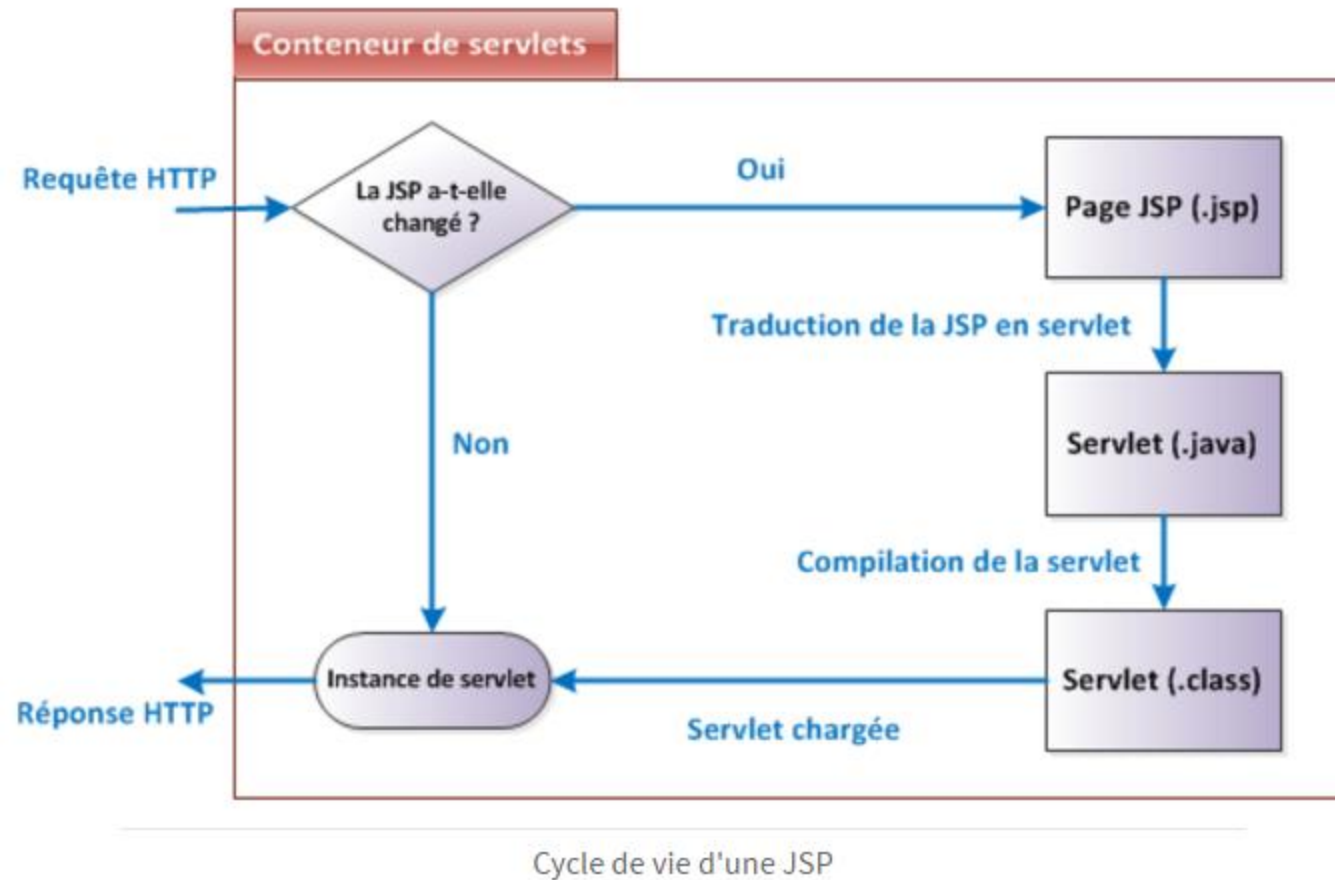
- **Traduction possible en Servlet:**

```
public void _jspService(HttpServletRequest req,  
                        HttpServletResponse res) throws ServletException, IOException {  
    res.setContentType("text/html");  
    JspWriter out=res.getWriter();  
    out.println( "<H1>Aujourd'hui nous sommes le </H1>");  
    out.println(new java.util.Date());  
    .....  
}
```

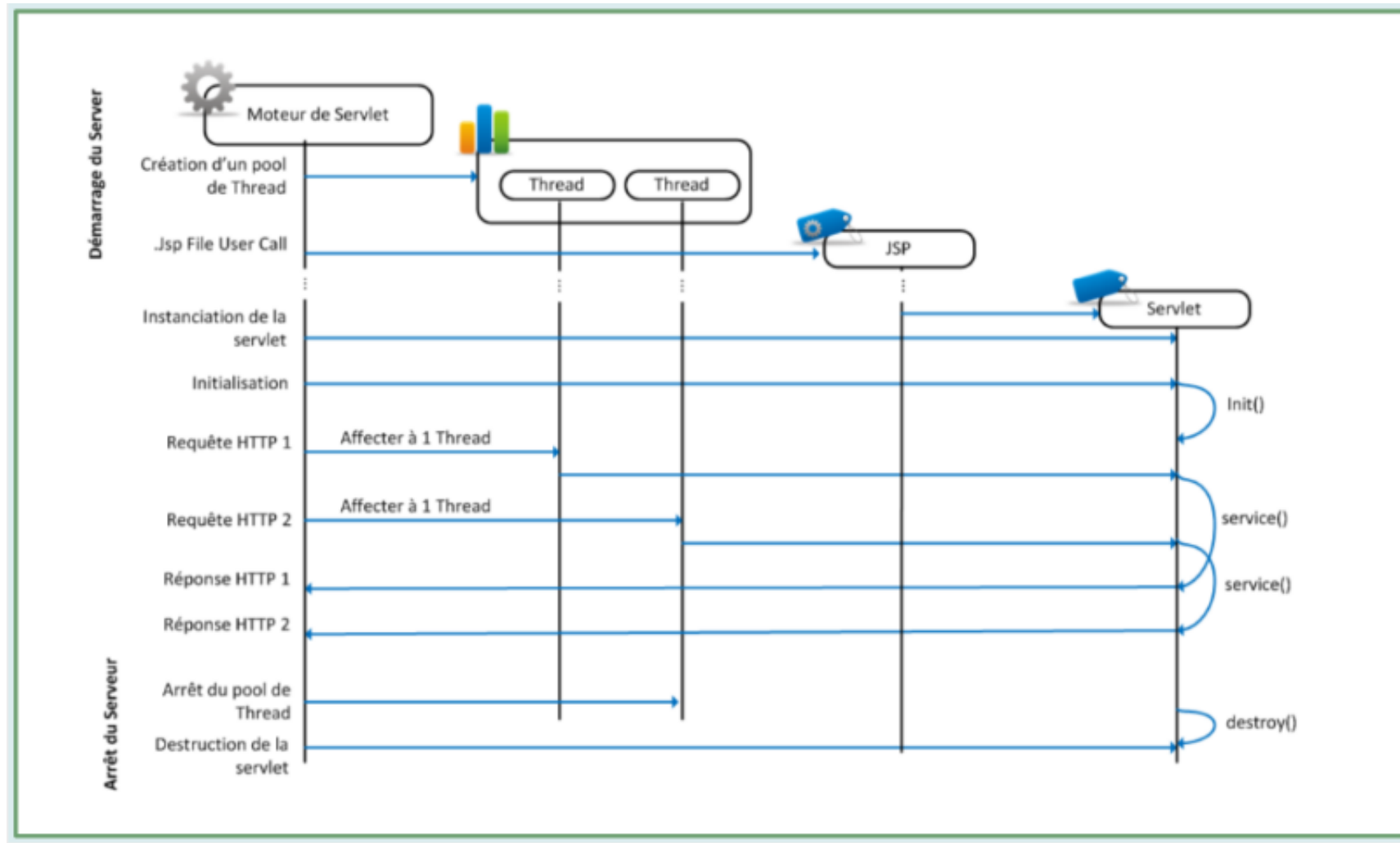

Cycle de vie d'une JSP (1/3)

- Lors de la réception d'une requête HTTP destinée à une JSP, le conteneur de servlets:
 - ✓ **vérifie si la JSP a déjà été traduite et compilée en une servlet:**
 - si non, il vérifie la syntaxe de la page, la traduit en une servlet (du code Java) et la compile en une classe exécutable prête à l'emploi ;
 - si oui, il vérifie que l'âge de la JSP et de la servlet est identique :
 - si non, cela signifie que la JSP est plus récente que la servlet et donc qu'il y a eu modification, le conteneur effectue alors à nouveau les tâches de vérification, traduction et compilation ;
 - ✓ **charge ensuite la classe générée, en crée une instance et l'exécute pour traiter la requête.**

Cycle de vie d'une JSP (2/3)



Cycle de vie d'une JSP (3/3)



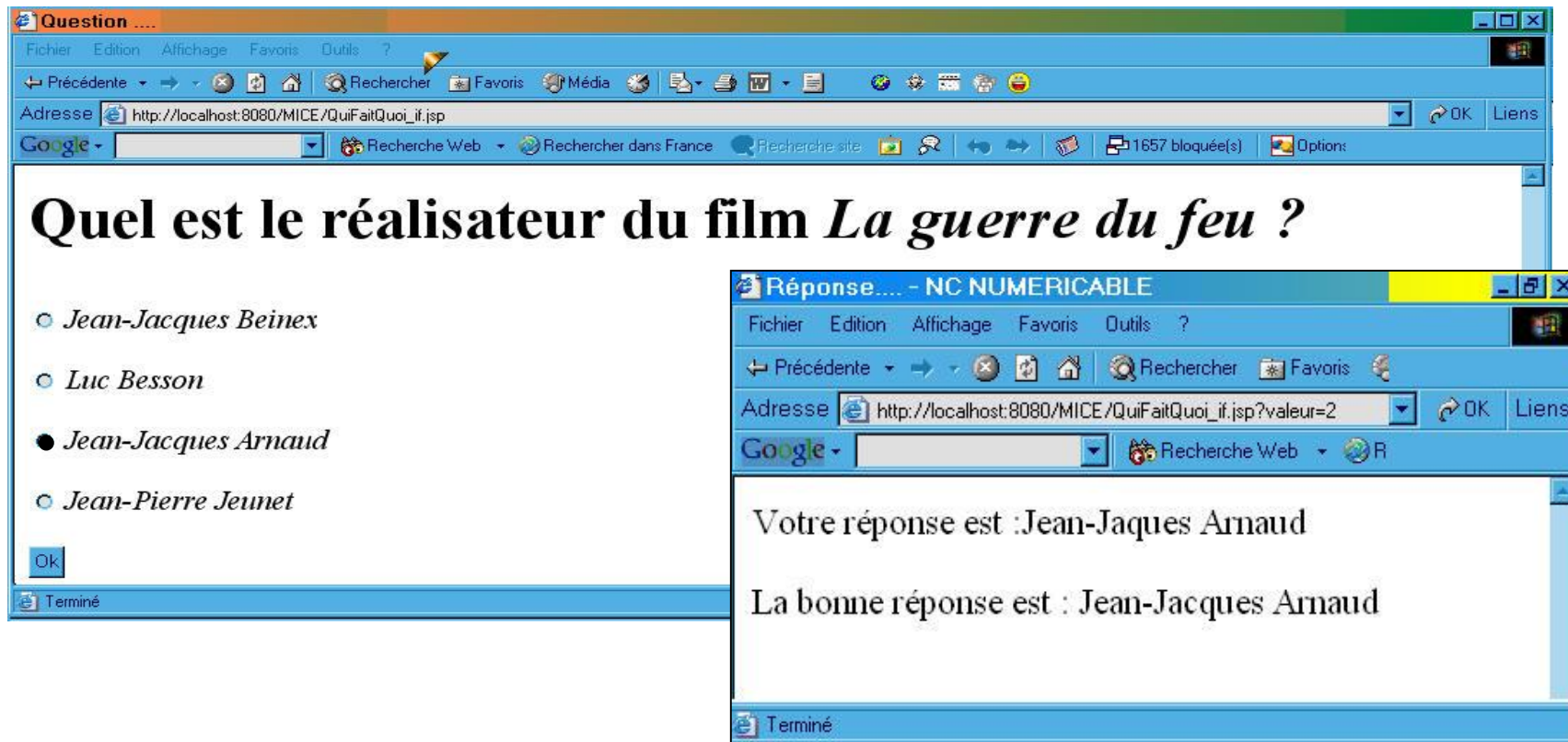
JSP et la programmation Java (1/10)

- **Ecrire un programme JSP qui se comporte différemment en fonction des paramètres qu'il récupère depuis l'URL d'appel du programme.**
 - ✓ Le premier appel du programme JSP s'effectue en lançant une requête par l'intermédiaire d'un URL simple, n'utilisant aucun paramètre, comme:
<http://localhost:8080/MaWebApp/MaJSP.jsp>
 - ✓ Le navigateur affiche en réponse un formulaire demandant à l'utilisateur de remplir le formulaire, puis valider son choix en cliquant sur le bouton OK.

JSP et la programmation Java (2/10)

- Exemple1: Programme JSP appelé **QuiFaitQuoi_if.jsp**:

- ✗ Formulaire avec bouton radio,
- ✗ Manipulation de la structure **Java** if-else.



JSP et la programmation Java (3/10)

- Exemple1 (suite):

```
QuiFaitQuoi_if.jsp

<%@page import="java.util.*"%>
<!-- Script JSP Créé le : <%=new Date()%>-->
<html><head>
<%!
    String lRealisateur [] = {"Jean-Jaques Beineix","Luc Besson", "Jean-Jaques Arnaud", "Jean-Pierre Jeunet" };
    int i;
%>
<% String parametre = request.getParameter("valeur");
    if (parametre == null) {
        out.println("<TITLE> Question ....</TITLE>");
        out.println("</head>");
        out.println("<Body bgcolor=white <BR>");
    }
%>

    <!-- Début du formulaire html avec bouton radio -->
    <form name=ques action=http://localhost:8080/MICE/QuiFaitQuoi_if.jsp method=Get>
    <p><h1> Quel est le réalisateur du film <i> La guerre du feu </i>?</h1><BR></p>
    <input type="Radio" name="valeur" value="0"> Jean-Jacques Beinex <BR><BR>
    <input type="Radio" name="valeur" value="1"> Luc Besson <BR><BR>
    <input type="Radio" name="valeur" value="2"> Jean-Jacques Arnaud <BR><BR>
    <input type="Radio" name="valeur" value="3"> Jean-Pierre Jeunet <BR><BR>
    <input type="submit" value="Ok"> </p>
    </form>

    <!-- Fin du formulaire html -->
```

JSP et la programmation Java (4/10)

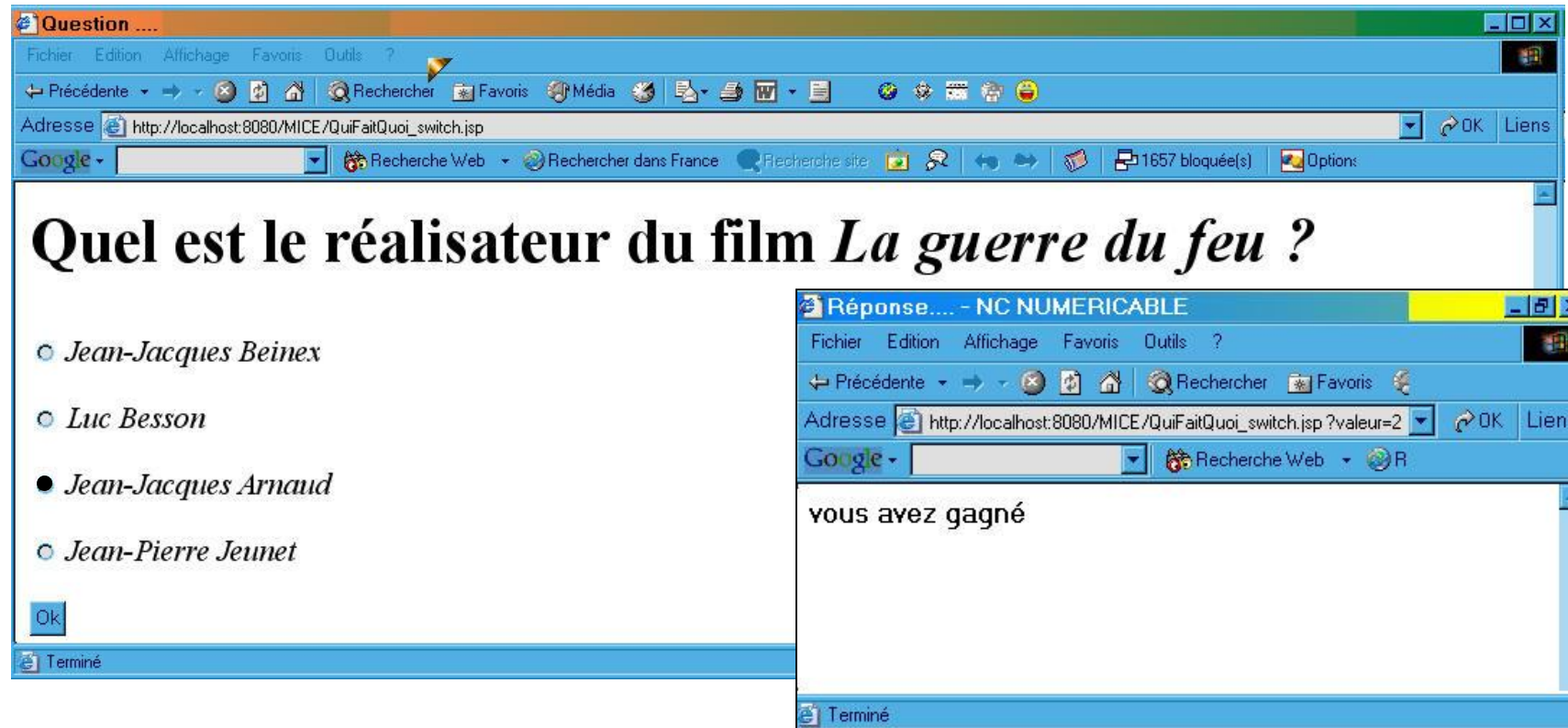
- **Exemple1 (suite):**

```
QuiFaitQuoi_if.jsp (suite)
<%
}
else {
    i= Integer.parseInt(parametre);
    out.println("<Title> Réponse.....</TITLE>");
    out.println("</head>");
    out.println("<Body bgcolor=white <BR>");
    out.println("Votre réponse est :"+ IRealisateur[i]+"<BR><BR>");
    out.println("La bonne réponse est : Jean-Jacques Arnaud");
}
%>
</head></html>
```

JSP et la programmation Java (5/10)

- **Exemple2: Programme JSP appelé `QuiFaitQuoi_switch.jsp`:**

- ✗ Formulaire avec bouton radio,
- ✗ Manipulation de la structure `Java switch`.



JSP et la programmation Java (6/10)

- Exemple2 (suite):

```
QuiFaitQuoi_switch.jsp

<%@page import="java.util.*"%>
<!-- Script JSP Créé le : <%=new Date()%>-->
<html><head>
<%!
    String lRealisateur [] = {"Jean-Jaques Beineix","Luc Besson", "Jean-Jaques Arnaud", "Jean-Pierre Jeunet" };
    int i;
%>
<% String parametre = request.getParameter("valeur");
    if (parametre == null) {
        out.println("<TITLE> Question ....</TITLE>");
        out.println("</head>");
        out.println("<Body bgcolor=white <BR>");
%>
    <!-- Début du formulaire html avec bouton radio -->
    <form name=ques action=http://localhost:8080/MICE/QuiFaitQuoi_switch.jsp method=Get>
    <p><h1> Quel est le réalisateur du film <i> La guerre du feu </i>?</h1><BR></p>
    <input type="Radio" name="valeur" value="0"> Jean-Jacques Beinex <BR><BR>
    <input type="Radio" name="valeur" value="1"> Luc Besson <BR><BR>
    <input type="Radio" name="valeur" value="2"> Jean-Jacques Arnaud <BR><BR>
    <input type="Radio" name="valeur" value="3"> Jean-Pierre Jeunet <BR><BR>
    <input type="submit" value="Ok"> </p>
    </form>
    <!-- Fin du formulaire html -->
```

JSP et la programmation Java (7/10)

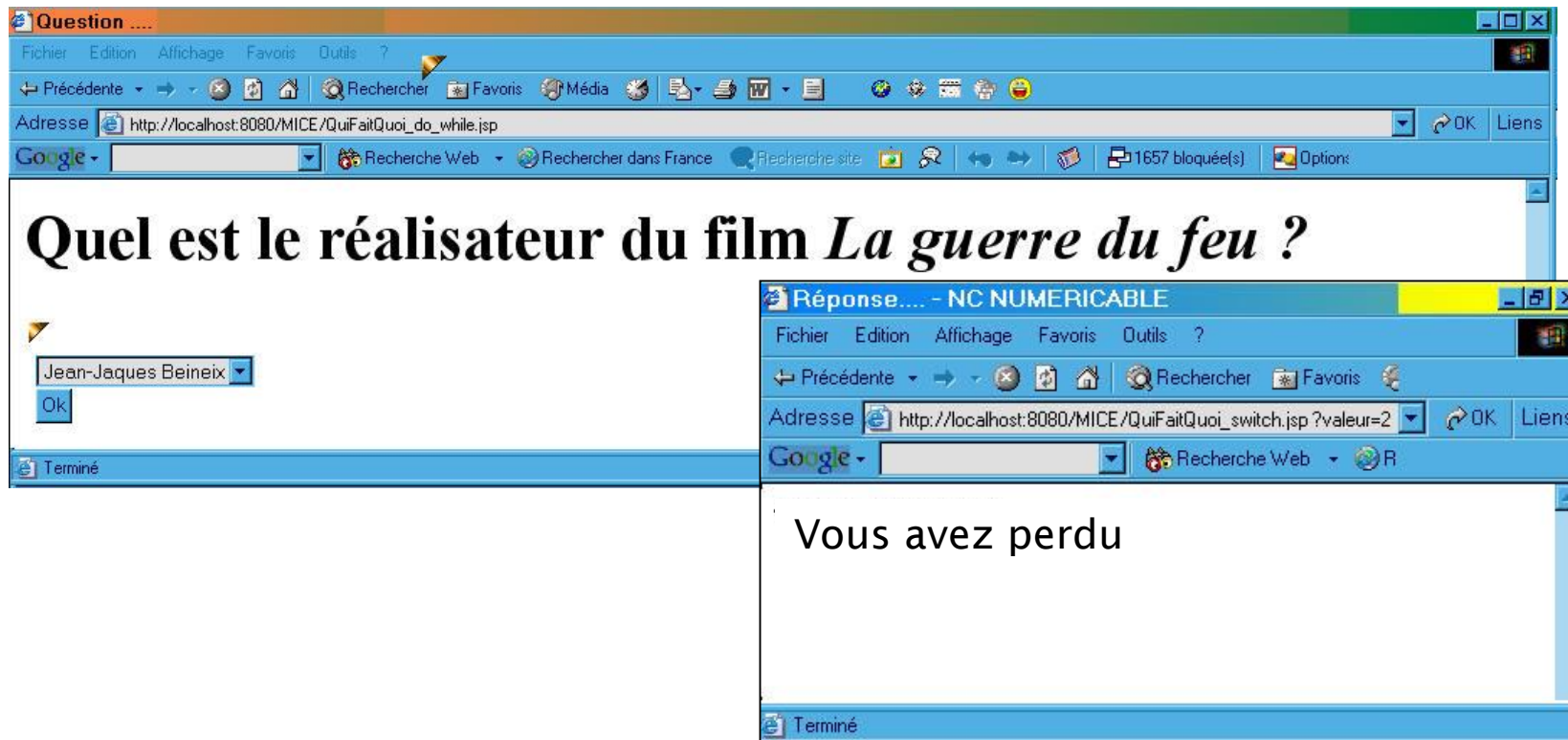
- **Exemple2 (suite):**

```
QuiFaitQuoi_switch.jsp (suite)
<%
}
else {
    i= Integer.parseInt(parametre);
    out.println("<Title> Réponse.....</TITLE>");
    out.println("</head>");
    out.println("<Body bgcolor=white <BR>");
    switch (i) {
        case 0 : case 1: case 3 : out.println("vous avez perdu ! ");
                    break;
        case 2 : out.println("vous avez gagnez ! ");
                    break;
    }
}
%>
</head></html>
```

JSP et la programmation Java (8/10)

- **Exemple3: Programme JSP appelé `QuiFaitQuoi_do_while.jsp`:**

- ✗ Formulaire avec liste de choix,
- ✗ Manipulation de la structure `Java` `do while`.



JSP et la programmation Java (9/10)

- Exemple3 (suite):

```
QuiFaitQuoi_do_while.jsp

<%@page import="java.util.*"%>
<!-- Script JSP Créé le : <%=new Date()%>-->
<html><head>
<%!
    String lRealisateur [] = {"Jean-Jaques Beineix","Luc Besson", "Jean-Jaques Arnaud", "Jean-Pierre Jeunet" };
    int i=0;
%>
<%
    String parametre = request.getParameter("auteur");
    if (parametre == null) {
        out.println("<TITLE> Question ....</TITLE>");
        out.println("</head>");
        out.println("<Body bgcolor=white <BR>");
    }
%>
<!-- Début du formulaire html avec bouton radio -->
<form name=ques action=http://localhost:8080/MICE/QuiFaitQuoi_do_while.jsp method=Get>
<h1> Quel est le réalisateur du film <i> La guerre du feu </i>?</h1><BR>
<%
    String chaineR="< select name=auteur>\n";
    do { chaineR+="<option value=" +i+">" +lRealisateur[i]+"</option>\n";
        i++;
    } while (i<lRealisateur.length);
    chaineR+="</select>";
%>
```

JSP et la programmation Java (10/10)

- **Exemple3 (suite):**

```
QuiFaitQuoi_switch.jsp (suite)
<%=chaineR%>
<input type="submit" value="Ok">
</form>
<!-- Fin du formulaire html -->
<%
    }
    else {
        int j= Integer.parseInt(parametre);
        out.println("<Title> Réponse....</TITLE>");
        out.println("</head>");
        out.println("<Body bgcolor=white <BR>");
        if(j==2) outprintln("vous avez gagné");
        else out.println("vous avez perdu");
    }
%>
</head></html>
```

Les actions

- Syntaxe XML.
- Les actions servent à transférer le contrôle d'une page à une autre.
- Elles permettent d'utiliser des JavaBeans.
- Les actions standard sont:
 - ✓ `<jsp:include>`
 - ✓ `<jsp:forward>`
 - ✓ `<jsp:useBean>`
 - ✓ `<jsp:getProperty>`
 - ✓ `<jsp:setProperty>`

L'action <jsp:include>

- Elle permet d'inclure un fichier au moment de la requête.
- Donc si le fichier inclus a été modifié, c'est la nouvelle version qui est incluse.

<jsp:include page="autre.jsp" />

- On peut passer des paramètres à la page incluse.

<jsp:include page="autre.jsp" >

<jsp:param name="user" value="michel" />

</jsp:include>

L'action <jsp:forward>

- Cette action, comme on s'y attend, effectue un forward vers une autre page.

<jsp:forward page="urlRelative" />

- Comme pour <jsp:include>, on peut préciser des paramètres supplémentaires à transmettre :

```
<jsp:forward page="login.jsp" >  
    <jsp:param name="user" value="moi"/>  
    <jsp:param name="password" value="xyz"/>  
</jsp:forward>
```


L'action <jsp:useBean>

- L'utilisation de javabeans est la première façon de limiter le code Java dans la page jsp.
- Un javabean doit avoir:
 - ✓ un constructeur sans paramètre
 - ✓ pas de champs public
 - ✓ un accès aux propriétés via des getters et setters get**Xxx**/set**Xxx** pour une propriété **xxx**.

<jsp:useBean id="nom" class="..." scope="..." />

L'action <jsp:getProperty>

- Quand on utilise un Javabeau, cette action permet d'obtenir la valeur de la propriété demandée.

<jsp:geProperty name="nomBean" property="xxx" />

- Cette action a le même effet que

<%= nomBean.getXxx() %>

- Elle contribue encore à supprimer le code Java dans la page JSP.

L'action <jsp:setProperty>

- Cette action permet de modifier une propriété du Javabean utilisé.
- Elle admet différents syntaxes:
 - ✓ **<jsp:setProperty name="nomDuBean" property="xxx" value="valeur" />**
a le même effet que
<% nomDuBean.setXxx("valeur"); %>
 - ✓ **<jsp:setProperty name="nom" property="xxx" param="nomParam"/>**
associe automatiquement la valeur du paramètre indiqué comme valeur de la propriété xxx du bean. Ceci a le même effet que
<% nom.setXxx(request.getParameter("nomParam"));%>

Example (1/3)

JavaBean

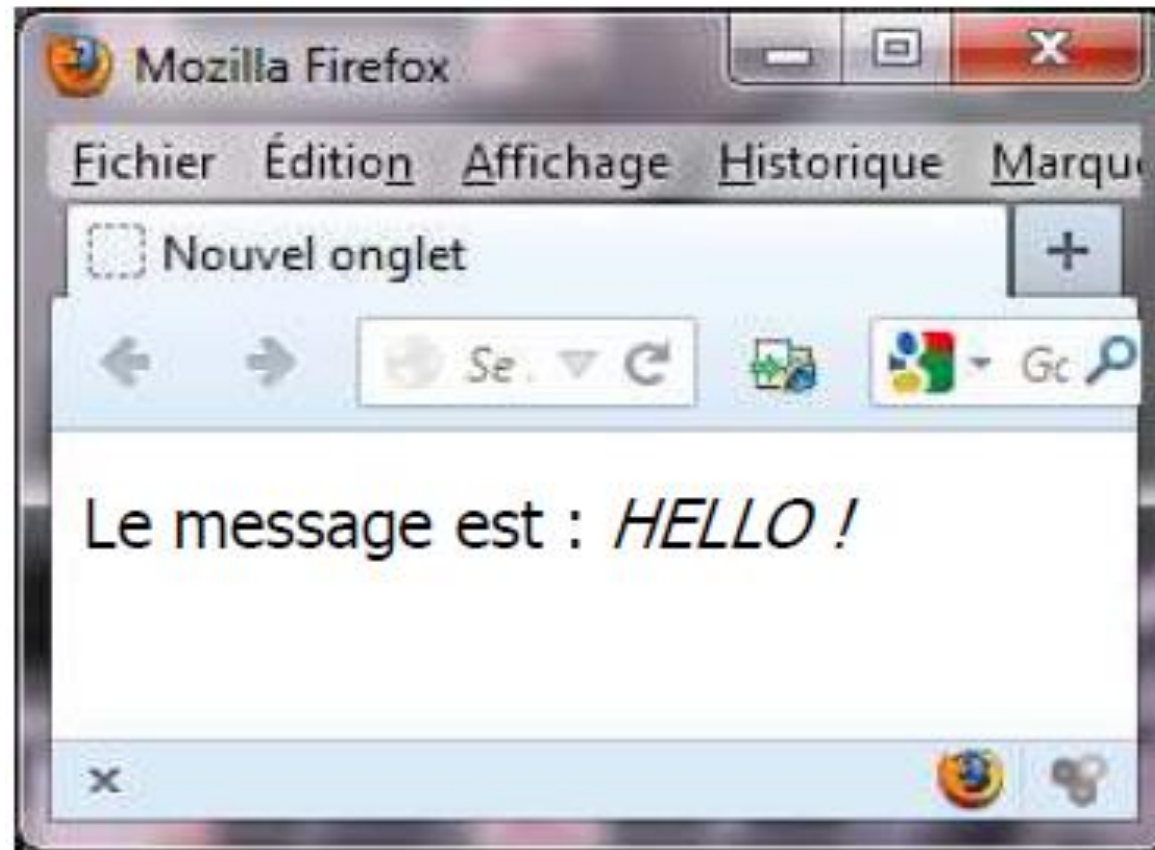
```
public class SimpleBean {  
    private String message = "";  
  
    public void setMessage(String msg) {  
        message = msg;  
    }  
  
    public String getMessage() {  
        message= SimpleBean();  
        return message;  
    }  
  
    private String SimpleBean(){  
        return message.toUpperCase()  
    }  
}
```

Exemple (2/3)

Page JSP

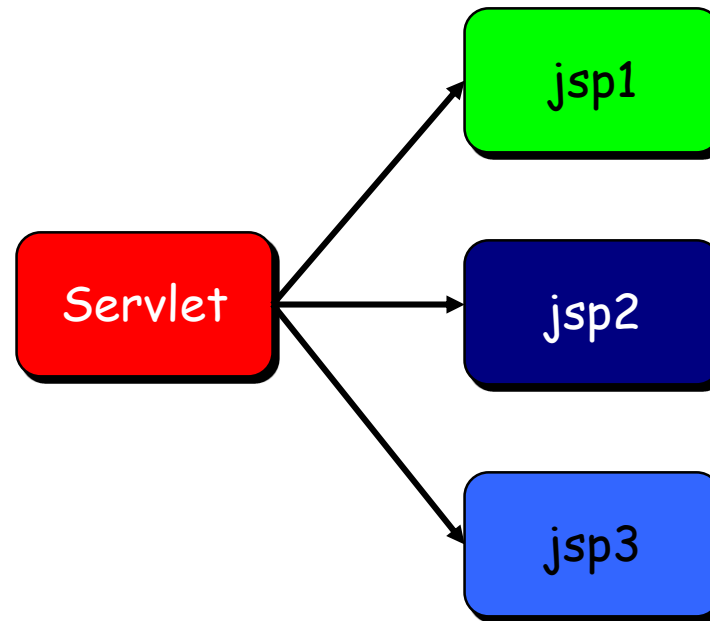
```
<html>
<body>
<jsp:useBean id="test" class="SimpleBean" />
<jsp:setProperty name="test" property="message" value="hello !" />
<p>Le message est :
<i><jsp:getProperty name="test" property="message" /></i></p>
</body>
</html>
```

Exemple (3/3)



Collaboration entre JSP et Servlets: L'Interface RequestDispatcher (1/2)

- **Agrégation de traitements fournis par des JSP:**
 - ✓ meilleure modularité,
 - ✓ meilleure réutilisation.



Collaboration entre JSP et Servlets: L'Interface RequestDispatcher (2/2)

- **Obtention d'un RequestDispatcher :**

- ✓ dans la méthode de traitement de requête de Servlet

```
.....  
RequestDispatcher rd;  
rd = getServletContext().getRequestDispatcher("/*****.jsp ");  
if(rd==null) res.sendError(404);  
.....
```

- **Redirection d'une requête:**

- ✓ dans la méthode de traitement de requête, demande à un JSP de répondre au client

```
rd.forward(req, res);
```